ACES-JS项目提交文档

1. 项目介绍

- ACES-JS项目是基于Restful接口规范开发的一款具有丰富样式和功能实现的Web端短视频应用，实现基础的视频播放/切换/切换过渡动画/分类功能/视频预加载，以及进阶的账户系统（注册/登录/注销/修改/历史记录/消息通知/视频上传/视频删除/视频置顶）、推荐系统（用户侧个性化推荐/视频侧相关推荐/视频推流加热/流量充值/广告投放）、交互系统（点赞/分享/关注/评论/弹幕/分享/搜索）等多达五十余项功能、设计大小不等的共十几个界面。
- 项目代码地址：https://github.com/LabEnbug/ACES-JS/
- 项目实例访问地址：http://101.133.129.34/

2. 项目分工

团队成员	主要贡献
张桓	负责数据库设计，后端接口设计开发，前端用户页，搜索页，充值，购买流量等页面开发，爬取测试视频。文档撰写。
万成全	负责前端短视频页面开发，互动界面开发，评论功能开发，弹幕功能开发。文档撰写。
蒋涛	负责推荐算法开大，推荐算法嵌入，文档撰写，项目部署。

3. 项目实现

3.1 技术选型

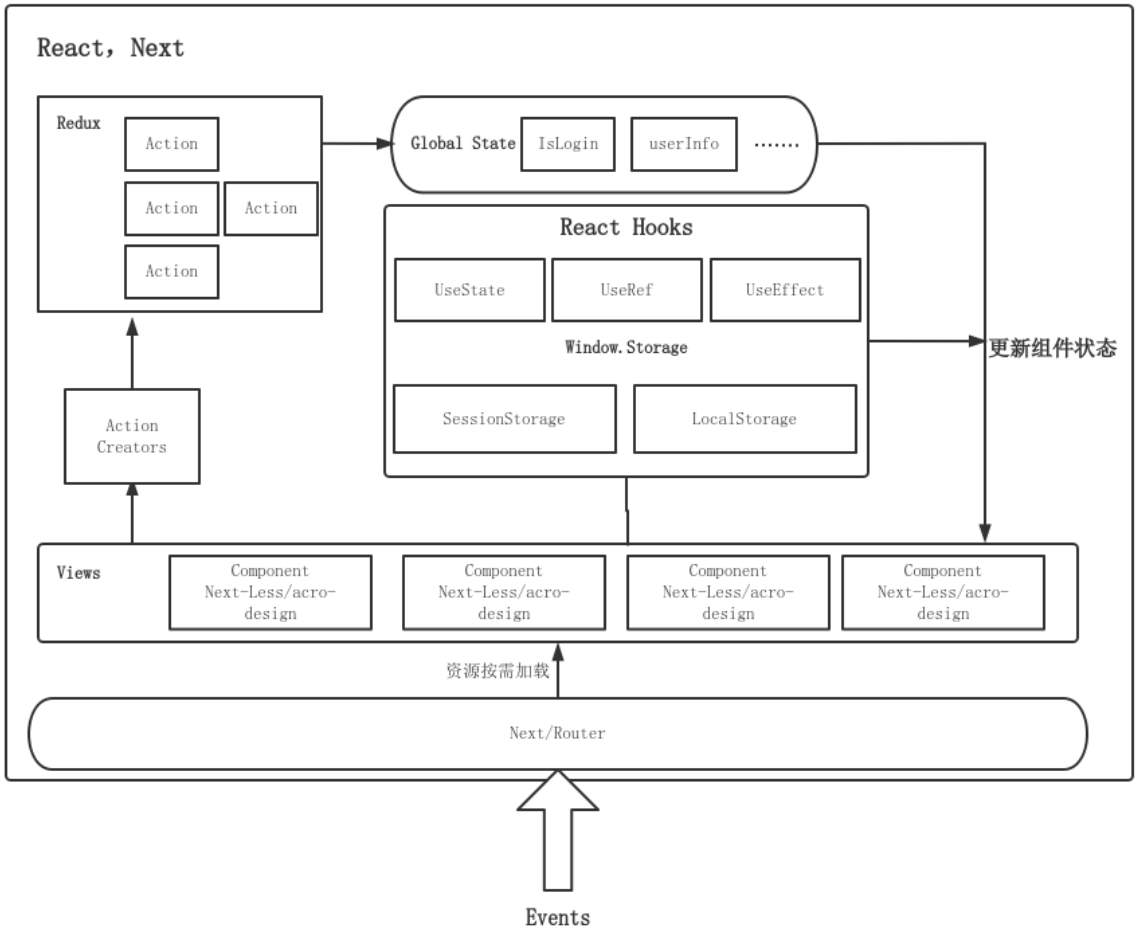
- Go：Goroutine使得并发和并行任务变得简单易行，为高效处理HTTP API访问、多个视频上传、下载和转码过程提供有效帮助。轻量级和高性能特性为后期构建微服务提供帮助（前期考虑，后期没有时间做服务切分）。
- Redis：用于存储token数据，便于管理用户同时登录数，也便于需要的时候将用户登录状态注销。
- Bearer token认证/JWT：无状态验证，不需要维护session，大幅减轻服务器的存储压力，减少高流量的短视频网站的服务器负担。无状态性可以在不同服务器或服务之间分散请求，不需要担心用户登录状态。支持跨域请求。简单和易于集成性有助于本项目的快速开发。
- Restful Api：无状态的Bearer配合使用，减少服务器的存储负担。具有统一的Api命名风格，有助于项目文档的规范化，前后端的沟通。
- MySQL Trigger：用于保证数据库数据完整性和一致性。
- React：著名的前端框架，利用虚拟DOM的机制，通过差异化更新的方式减少了对实际DOM的操作，有助于提升前端的性能表现，其函数式编程思维，和组件通过hook函数纯函数化，有助于代码风格的进步，且易于解耦，理解和维护。
- React-redux：Redux提供了保护管理状态一致性的容器，帮助维护需要多组件共享的状态（如用户登录状态，用户信息等）的一致性。
- axios：更快速、高效地开发和维护与后端服务的交互，同时为用户提供快速和可靠的视频内容加载体验。
- Next.js：服务端渲染能力可以提高短视频网站的首屏加载性能。内置的 API 路由无需额外创建路由文件。Next.js 自动的代码拆分，极大提升页面加载速度，有利于需要快速渲染视频内容的短视频网站的访问效率。
- HLS视频流：HLS视频流通过将视频文件切分成数秒到十几秒的小块文件，可以提高用户访问视频时的加载速度。
- arco-design/icon-park：提供经过验证的前端组件/图标，保证前端风格一致性，提升开发效率。
- Video.js：提供支持HLS的视频播放功能，有助于视频样式的自定义，视频功能拓展。
- less：高效编写和维护样式表。
- 七牛云：用于存储静态资源如视频、封面图片、用户头像，配合CDN的使用使得用户在何处都可以低延迟访问资源；加入云转码功能，可以使得用户上传的短视频在存储到七牛云时即可触发进入转码队列，回调的使用可以让短视频网站获得转码成功的通知。
- 基于动态概率矩阵的推荐算法：根据用户的观看历史、点赞、评论和分享等行为构建动态概率矩阵，为用户提供个性化推荐；实时更新用户的兴趣模型、调整推荐策略；引入一定的随机性，避免信息茧房；可解释性强，推荐结果与相应的概率关联，有利于增加透明度和提升用户信任度。
- JSX：结合react，实现在js代码里嵌入html代码。
- TypeScript：结合类型语言的优点，令代码编写更为规范化，帮助更快定位问题所在。

平台部署在一台阿里云服务器上，系统为Debian 12。

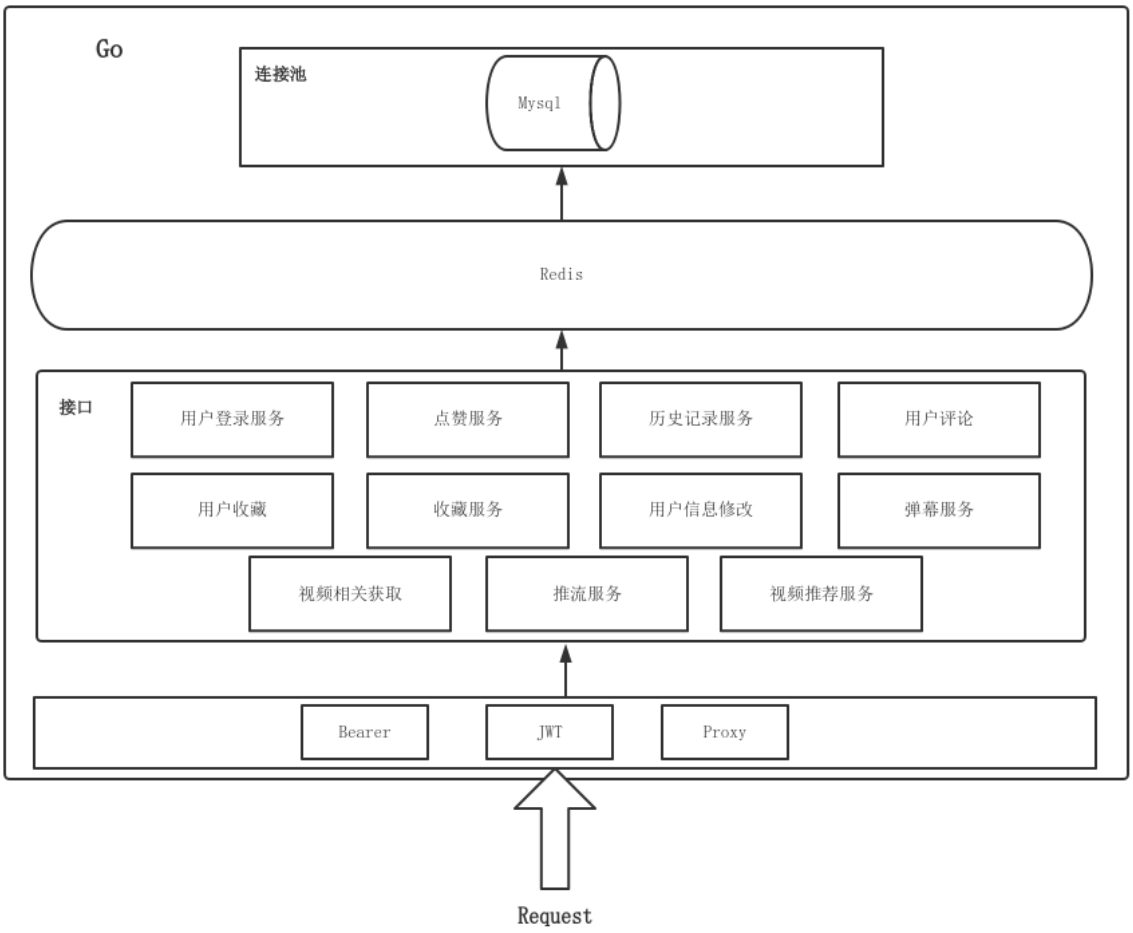
3.2 架构设计

本项目实行前后端分离的策略，并易于拆分成多个微服务缓解服务器请求的压力。

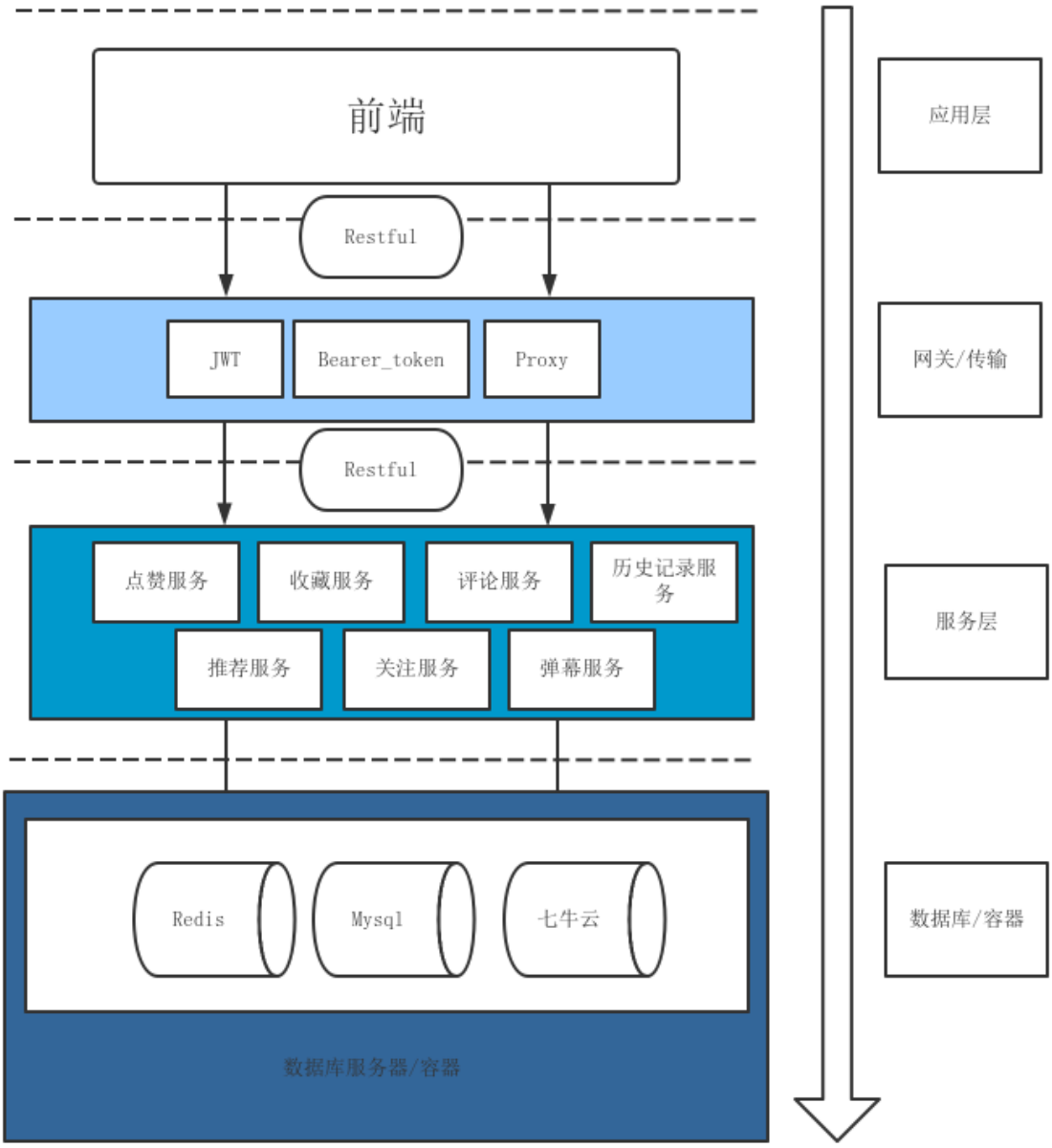
3.2.1 前端架构



3.2.2 后端架构


Go
连接池
Mysql
Redis
接口
用户登录服务
点赞服务
历史记录服务
用户评论
用户收藏
收藏服务
用户信息修改
弹幕服务
视频相关获取
推流服务
视频推荐服务
Bearer
JWT
Proxy
Request

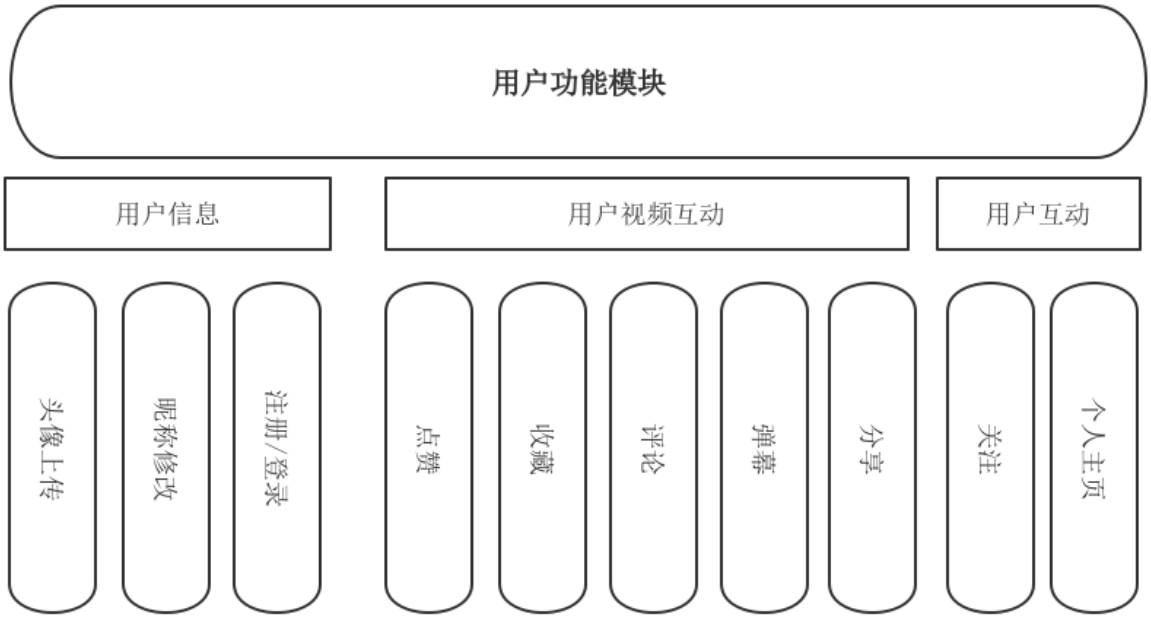
3.2.3 整体架构



3.3 模块及其功能介绍

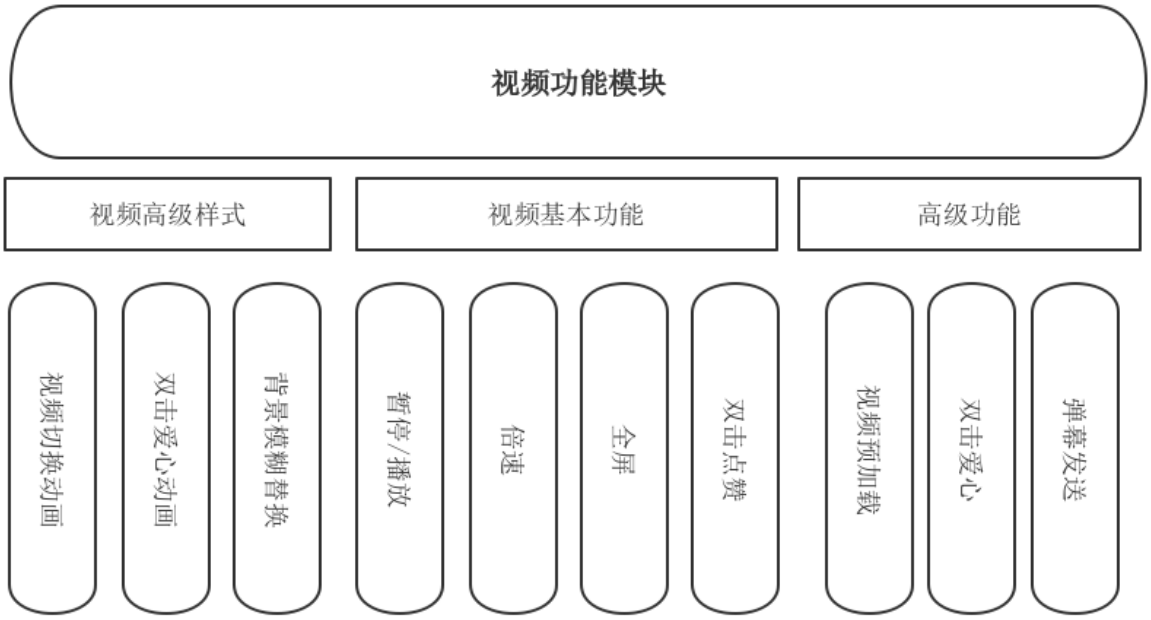
3.3.1 用户模块

用户模块集成了用户信息修改，点赞，弹幕发送，评论，信息修改等一系列功能。



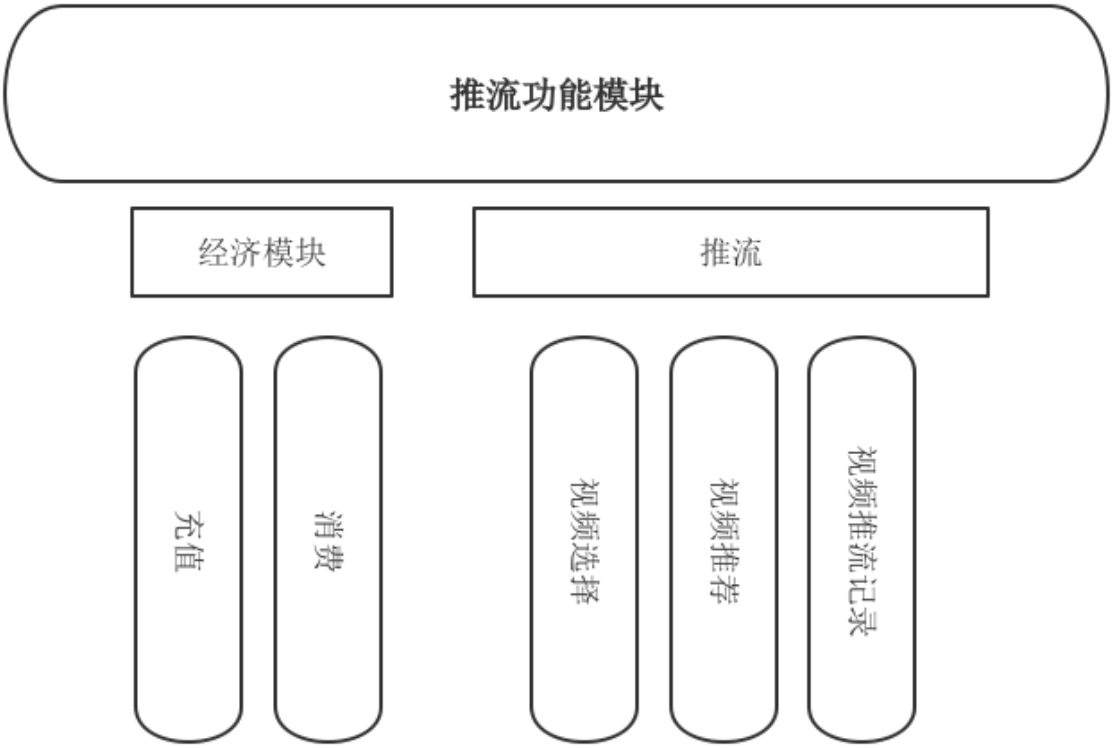
3.3.2 视频功能模块

视频功能模块，集成了弹幕，双击点赞，相关视频推荐，暂停，全屏，音量修改，倍速修改，连播等功能。



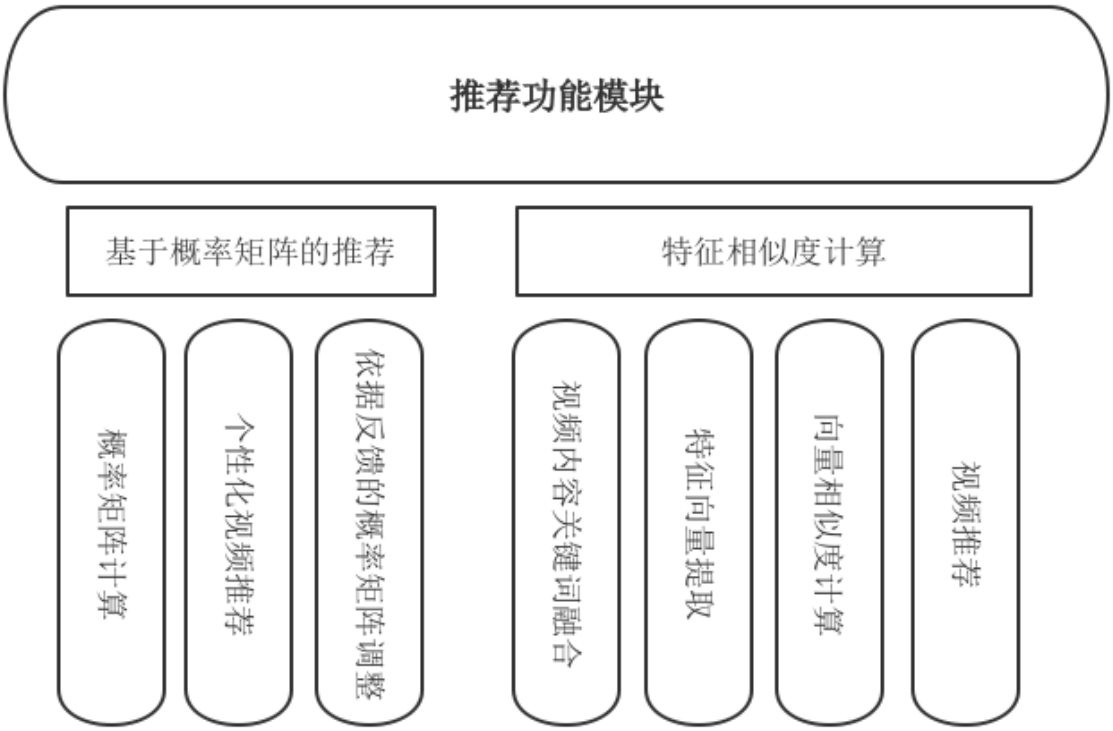
3.3.3 推流功能模块

推流功能模块集成了推流余额，推流视频选择等功能，支持用户为视频充值流量，系统将自动推荐充值次数的视频给关注。例如，博主A花费10金币为自己刚刚上传的视频购买了100流量，短视频平台会将此视频推荐给观众，共计100次，完成这100次推广后，该视频将回归正常的被观看逻辑。



3.3.4 推荐功能模块

推荐功能模块分为用户侧的个性化推荐和视频侧的基于视频内容和关键词的相关视频推荐，多方位为用户推荐合适的视频。



3.4 项目代码介绍

3.4.1 代码目录说明

```
🙂 ├── acro_frontend #前端目录
   |   ├── next.config.js
   |   ├── next-env.d.ts
   |   ├── package.json
   |   ├── README.md
   |   ├── src #源码目录
   |   |   ├── components #通用组件目录
   |   |   |   ├── Footer #脚注组件
   |   |   |   |   ├── index.tsx
   |   |   |   |   └── style
   |   |   |   |       └── index.module.less
   |   |   |   ├── MessageBox #信息通知
   |   |   |   |   ├── index.tsx
   |   |   |   |   ├── list.tsx
   |   |   |   |   └── style
   |   |   |   |       └── index.module.less
   |   |   |   ├── NavBar # 页面顶部组件
   |   |   |   |   ├── IconButton.tsx
   |   |   |   |   ├── index.tsx
   |   |   |   |   └── style
   |   |   |   |       ├── icon-button.module.less
   |   |   |   |       └── index.module.less
   |   |   |   ├── Panel
   |   |   |   |   ├── index.tsx
   |   |   |   |   └── style
   |   |   |   |       └── index.module.less
   |   |   |   ├── PermissionWrapper #权限管理
   |   |   |   |   └── index.tsx
   |   |   |   ├── SearchPopupBox #搜索框
   |   |   |   |   ├── index.tsx
   |   |   |   |   └── style
   |   |   |   |       └── index.module.less
   |   |   |   ├── Settings #设置
   |   |   |   |   ├── block.tsx
   |   |   |   |   ├── color.tsx
   |   |   |   |   ├── index.tsx
```

```
|   |   |   |   └── style
|   |   |   |       ├── block.module.less
|   |   |   |       └── color-panel.module.less
|   |   |   ├── SiderTabs  #右边弹出框
|   |   |   |   ├── Comment #评论相关组件目录
|   |   |   |   |   ├── index.tsx
|   |   |   |   |   ├── locale
|   |   |   |   |   |   └── index.ts
|   |   |   |   |   ├── replay.tsx
|   |   |   |   |   └── style
|   |   |   |   |       └── index.module.less
|   |   |   |   ├── index.tsx
|   |   |   |   └── locale
|   |   |   |       └── index.ts
|   |   |   └── Video #视频相关组件目录
|   |   |       ├── brief_intro.tsx
|   |   |       ├── footbar.tsx
|   |   |       ├── index.tsx
|   |   |       ├── locale
|   |   |       |   └── index.ts
|   |   |       ├── sidebar.tsx
|   |   |       └── style
|   |   |           ├── index.module.less
|   |   |           └── intro.module.less
|   |   ├── context.tsx
|   |   ├── declaration.d.ts
|   |   ├── locale #本地化目录
|   |   |   └── index.ts
|   |   ├── pages #页面目录
|   |   |   ├── advertise #推广页面
|   |   |   |   ├── index.tsx
|   |   |   |   └── style
|   |   |   |       └── index.module.less
|   |   |   ├── _app.tsx
|   |   |   ├── deposit #余额页面
|   |   |   |   ├── index.tsx
|   |   |   |   └── style
```

```
|   |   |   |   └── index.module.less
|   |   |   ├── edit # 编辑页面
|   |   |   |   ├── index.tsx
|   |   |   |   └── style
|   |   |   |       └── index.module.less
|   |   |   ├── exception # 错误显示页面
|   |   |   |   ├── 403
|   |   |   |   |   ├── index.tsx
|   |   |   |   |   ├── locale
|   |   |   |   |   |   └── index.ts
|   |   |   |   |   └── style
|   |   |   |   |       └── index.module.less
|   |   |   |   ├── 404
|   |   |   |   |   ├── index.tsx
|   |   |   |   |   ├── locale
|   |   |   |   |   |   └── index.ts
|   |   |   |   |   └── style
|   |   |   |   |       └── index.module.less
|   |   |   |   └── 500
|   |   |   |       ├── index.tsx
|   |   |   |       ├── locale
|   |   |   |       |   └── index.ts
|   |   |   |       └── style
|   |   |   |           └── index.module.less
|   |   |   ├── index.tsx
|   |   |   ├── layout.tsx
|   |   |   ├── login #登录页面
|   |   |   |   ├── banner.tsx
|   |   |   |   ├── form.tsx
|   |   |   |   ├── index.tsx
|   |   |   |   ├── locale
|   |   |   |   |   └── index.ts
|   |   |   |   └── style
|   |   |   |       └── index.module.less
|   |   |   ├── promote #推广页面
|   |   |   |   ├── index.tsx
|   |   |   |   ├── locale
```

```
|   |   |   |   |   └── index.ts
|   |   |   |   └── style
|   |   |   |       └── index.module.less
|   |   |   ├── result #结果页面
|   |   |   |   ├── error
|   |   |   |   |   ├── index.tsx
|   |   |   |   |   ├── locale
|   |   |   |   |   |   └── index.ts
|   |   |   |   |   └── style
|   |   |   |   |       └── index.module.less
|   |   |   |   └── success
|   |   |   |       ├── index.tsx
|   |   |   |       ├── locale
|   |   |   |       |   └── index.ts
|   |   |   |       └── style
|   |   |   |           └── index.module.less
|   |   |   ├── search #搜索页面
|   |   |   |   ├── card-block.tsx
|   |   |   |   ├── index.tsx
|   |   |   |   ├── interface.ts
|   |   |   |   ├── locale
|   |   |   |   |   └── index.ts
|   |   |   |   └── style
|   |   |   |       └── index.module.less
|   |   |   ├── upload #上传页面
|   |   |   |   ├── index.tsx
|   |   |   |   └── style
|   |   |   |       └── index.module.less
|   |   |   ├── user #用户页面
|   |   |   |   ├── card-block.tsx
|   |   |   |   ├── interface.ts
|   |   |   |   ├── locale
|   |   |   |   |   └── index.ts
|   |   |   |   ├── style
|   |   |   |   |   └── index.module.less
|   |   |   |   ├── user-addon-count-info.tsx
|   |   |   |   └── [username].tsx
```

```
|   |   |   └── video #短视频页面
|   |   |       ├── index.tsx
|   |   |       ├── locale
|   |   |       |   └── index.ts
|   |   |       └── style
|   |   |           └── index.module.less
|   |   ├── public #静态资源
|   |   |   ├── assets
|   |   |   |   ├── dark.svg
|   |   |   |   ├── entertainment.svg
|   |   |   |   ├── fantasy.svg
|   |   |   |   ├── fashion.svg
|   |   |   |   ├── food.svg
|   |   |   |   ├── game.svg
|   |   |   |   ├── hot.svg
|   |   |   |   ├── knowledge.svg
|   |   |   |   ├── light.svg
|   |   |   |   ├── logo.svg
|   |   |   |   ├── payment-check.svg
|   |   |   |   ├── sport.svg
|   |   |   |   └── world.json
|   |   |   ├── favicon.ico
|   |   |   └── vercel.svg
|   |   ├── routes.ts # route文件
|   |   ├── settings.json
|   |   ├── store # store目录
|   |   |   └── index.ts
|   |   ├── style
|   |   |   ├── global.less
|   |   |   └── layout.module.less
|   |   └── utils # 通用函数目录
|   |       ├── authentication.ts
|   |       ├── changeTheme.ts
|   |       ├── clipboard.ts
|   |       ├── getaxios.tsx
|   |       ├── getUrlParams.ts
|   |       ├── getuserinfo.tsx
```

```
|   |   ├── getvideotype.ts
|   |   ├── is.ts
|   |   ├── keywordUtils.tsx
|   |   ├── lazyload.tsx
|   |   ├── setupMock.ts
|   |   ├── timeUtils.js
|   |   ├── useChartTheme.ts
|   |   ├── useLocale.ts
|   |   └── useStorage.ts
|   ├── tsconfig.json
|   └── yarn.lock
└── backend #后端目录
    ├── algorithm #推荐算法
    |   └── recommandation.go
    ├── auth #认证
    |   ├── ecdsa.go
    |   ├── ecdsa_private.pem
    |   ├── ecdsa_public.pem
    |   └── jwt.go
    ├── cmd #用户操作相关
    |   ├── callback.go
    |   ├── common.go
    |   ├── response.go
    |   ├── search.go
    |   ├── user.go
    |   ├── video_bullet_comment.go
    |   ├── video_comment.go
    |   └── video.go
    ├── common #公用函数
    |   └── showData.go
    ├── config #配置
    |   └── config.go
    ├── database #数据库相关
    |   ├── mysql
    |   |   ├── callback.go
    |   |   ├── mysql.go
    |   |   ├── recommend.go
```

```
|   |   ├── search.go
|   |   ├── user.go
|   |   ├── video_bullet_comment.go
|   |   ├── video_comment.go
|   |   └── video.go
|   └── redis.go
├── go.mod
├── go.sum
├── main.go
├── model
|   ├── qiniu_callback_data.go
|   ├── recommend.go
|   ├── user_deposit_card.go
|   ├── user.go
|   ├── video_action.go
|   ├── video_bullet_comment.go
|   ├── video_comment_child.go
|   ├── video_comment_root.go
|   ├── video_forward.go
|   ├── video.go
|   ├── video_history.go
|   └── video_type.go
├── README.md
└── tool
    ├── qiniu.go
    └── time.go
```

3.4.2 业务接口介绍

3.4.2.1 用户系统的接口介绍

1. **用户登录**

   获取前端传入的用户名和密码，认证后，返还Bearer Token

2. **用户注册**

获取前端传入的用户信息，校验后，返回登录是否成功和Bearer Token

3. **用户注销**

   根据传入信息，对用户进行注销，级联删除相关信息。

4. **获取用户信息**

   根据bearer token，返还用户的详细信息。

5. **修改用户昵称**

   通过bearer token认证，获取传入的昵称，校验后返回具体的操作结果。

6. **修改用户头像**

   通过bearer token认证，获取上传的头像信息后，返回具体的操作结果。

7. **获取其他用户信息**

   通过bearer token认证，返回其他用户的非私密信息。

8. **关注/取消关注用户**

   通过bearer token认证，和操作面对的用户id，返回关注/取消关注结果。

9. **充值**

   为用户账户充值。

3.4.2.2 视频获取的接口介绍

1. **获取单个视频信息**

   依据video_uid，返回视频信息（包括视频发布用户信息）。

2. **获取视频列表（无任何指定）**

   依据start，limit关键字，返回相应的视频列表信息。

3. **获取视频列表（指定类型）**

   依据start，limit，type关键字，返回相应的视频列表信息。

4. **获取视频列表（用户上传）**

   依据start，limit等关键字，返回相应的视频列表信息。

5. **获取视频列表（用户点赞过的）**

依据start，limit等关键字，返回相应的视频列表信息。

6. **获取视频列表（用户收藏的）**

   依据start，limit等关键字，返回相应的视频列表信息。

7. **获取视频列表（用户播放历史）**

   通过bearer token认证，依据start，limit等关键字，返回仅自己的播放历史信息。

8. **获取视频列表（用户关注博主的视频）**

   通过bearer token认证，依据start，limit，用户id等关键字，返回相应的视频列表信息。

9. **获取视频类型信息**

   依据type等关键字，返回视频类型信息。

10. **上传新视频**

    通过bearer token认证，上传视频，存储视频，返回上传结果。

11. **编辑视频信息**

    通过bearer token认证，视频内容介绍，关键词，视频类型等信息，修改视频相关信息，返回修改结果。

12. **删除视频**

    通过bearer token认证，依据video_uid等信息，删除视频，返回删除结果。

13. **确认发布新视频**

    补充视频信息发布视频。

14. **修改视频信息**

    根据输入字段修改视频信息。

3.4.2.3 交互系统的接口介绍

1. **记录播放历史**

   依据video_uid等关键字，返回记录播放历史结果。

2. **点赞**

   通过bearer token认证，依据video_uid等关键字，返回点赞结果。

3. **取消点赞**

通过bearer token认证，依据video_uid等关键字，返回取消点赞结果。

4. **收藏**

通过bearer token认证，依据video_uid等关键字，返回收藏结果。

5. **取消收藏**

通过bearer token认证，依据video_uid等关键字，返回取消收藏结果。

6. **评论**

通过bearer token认证，依据video_uid关键字，评论内容，引用的评论id等信息，返回评论发布结果。

7. **获取根评论/视频的根评论**

依据video_uid，start，limit等关键字，返回根评论列表，根评论中，最多携带1个子评论，其它需通过获取子评论来获取。

8. **获取子评论/视频的子评论**

依据根评论id，start，limit等关键字，返回子评论列表。

9. **删除评论**

通过bearer token认证，依据评论id关键字，删除评论，返回删除评论结果。

10. **发弹幕**

通过bearer token认证，依据video_uid，弹幕时间点关键字，存储弹幕，返回存储弹幕结果。

11. **获取弹幕列表**

依据video_uid，start，limit等关键字，获取弹幕列表，返回弹幕列表。

12. **删除弹幕**

通过bearer token认证，依据video_uid，弹幕uid，删除弹幕，返回弹幕删除结果。

13. **置顶视频**

在自己上传的视频页面设置置顶视频。

14. **设为私密**

在自己上传的视频页面设置某视频为私密视频

15. 推广视频

将自己上传的视频设为推广视频，为其充值流量。

16. 广告视频

设置某些视频为广告视频，为其充值广告的费用。

3.4.2.4 搜索功能的接口介绍

1. 视频搜索-获取hotkeys

依据max_count关键字，返回对应的热词列表。

2. 视频搜索

依据keyword，limit，start关键字，返回对应关键字的搜索结果。

3. 用户搜索

依据keyword，limit，start关键字，返回对应关键字的搜索结果。

4. 获取用户搜索hotkeys

依据max_count关键字，返回对应的热词列表。

3.4.2.5 推荐系统的接口介绍

1. 获取相关视频列表

依据当前播放的视频，利用视频侧的推荐算法，返回相关视频列表信息

2. 获取为用户推荐的视频列表

通过bearer token认证，依据用户的概率矩阵，利用用户侧的推荐算法，返回下滑时的视频列表信息

3.4.2.6 回调的接口介绍

1. 七牛视频转码回调

接收七牛云自动转码成功的回调信息，将视频标记为HLS视频切片可用（此时视频允许被观看）。

2. 七牛视频截图回调

接收七牛云自动截图成功的回调信息，将视频标记为封面可用。

3.4.3 Redis设计

本项目的Redis目前用于存储用户登录后的Token，设置3天的过期时间，同时限制最大登录session为2。

3.4.4 Mysql设计

3.4.4.1 数据库总体设计

本项目的数据库包含15个表，分别是：

1. 用户表 user：存储用户信息，包含概率推荐矩阵，密码存储使用bcrypt哈希加密，保证安全。
2. 用户充值表 user_deposit_card：存储用户充值卡，用于充值用户的账户余额。
3. 用户关注表 user_follow：存储用户关注的信息。
4. 视频表 video：存储视频信息，包含相关视频推荐列表。
5. 视频-广告表 video_advertise：存储用户的购买广告视频信息。
6. 视频-弹幕表 video_bullet_comment：存储用户发送在视频上的弹幕的信息。
7. 视频-评论表 video_comment：存储用户在视频下发送/回复/引用的评论的所有信息。
8. 视频-收藏表 video_favorite：存储用户收藏的视频。
9. 视频-分享表 video_forward：存储用户分享的视频。
10. 视频-喜欢表 video_like：存储用户喜欢的视频。
11. 视频-推流表 video_promote：存储用户为某视频购买的推广流量。
12. 视频-种类表 video_type：存储视频的种类。
13. 视频-观看表 video_watch：存储用户观看视频的信息。
14. 用户搜索历史表 search_history_user：存储用户的搜索历史记录，搜索用户。
15. 视频搜索历史表 search_history_video：存储用户的搜索历史记录，搜索视频。

3.3.4.2 触发器设计

为防止数据库的每个记录出现多一少一的错误，在各个数据表中设计总计15个触发器，分别是：

1. user_deposit_card_au：当充值卡被使用后，增加用户对应余额和充值总额。
2. user_follow_on：当有新粉丝点关注时，使关注数实现自动加一的更新。
3. user_follow_off：当有老粉丝取消关注时，使关注数实现自动减一的更新。
4. video_on：当视频被点赞、被收藏、被分享、被观看、被转发数加一时，使视频拥有者的被点赞、被收藏、被分享、被观看、被转发数加一。
5. video_advertise_ai：当有用户购买广告量时，使视频的广告剩余次数增加对应次数，并且减少用户购买广告花费的相应余额。
6. video_comment_on：当有用户评论时，使评论数实现自动加一的更新。
7. video_comment_off：当有用户删除评论时，使评论数实现自动减一的更新。

8. video_favorite_on：当有用户为视频点收藏时，使收藏数实现自动加一的更新。
9. video_favorite_off：当有用户为视频取消收藏时，使收藏数实现自动减一的更新。
10. video_forward_on：当有用户为视频点分享时，使分享数实现自动加一的更新。
11. video_like_on：当有用户为视频点喜欢时，使喜欢数实现自动加一的更新。
12. video_like_off：当有用户为视频取消喜欢时，使喜欢数实现自动减一的更新。
13. video_promote_ai：当有用户购买推广流量时，使视频的推广流量剩余次数增加对应次数，并且减少用户购买推广流量的相应余额。
14. video_watch_bi：当有用户观看视频时，当该视频仍有推广次数，则将本次观看置为是推广的视频。如果没有推广次数，并当该视频仍有广告次数，则将本次观看置为是广告视频。
15. video_watch_on：当本次观看是记录为推广视频时，扣除最老的一条购买推广流量记录的次数，并且同时扣除视频的剩余推广数，还要增加一次视频的被播放次数；当本次观看是记录为推广视频时，扣除最老的一条购买推广流量记录的次数，并且同时扣除视频的剩余推广数，还要增加一次视频的被播放次数；如果都不是，则只增加一次视频的被播放次数。

3.3.4.3 索引和外键设计

数据表的索引和外键的合理设计有助于优化查询、保证数据一致性等，因此，本项目在设计数据库时，考虑了索引和外键设计，具体有：

1. 用户关注表中，将user_id加入索引中，提升查询效率
2. 视频表中，将视频简介和关键词加入索引，帮助提高搜索功能的速度；另外，还将视频上传者的id加入索引并设置为外键，帮助提升跳转作者页面时返回其上传视频的速度。
3. 视频-广告表中，考虑到往往需要查询某个用户的所有广告信息，因此，将用户的id加入索引并设置为外键。
4. 视频-评论表中，从评论设计到的功能来说：首先，查看在某视频下的评论，需要查询当前视频对应的所有评论，即将视频id加入索引以加速此查询；其次，在某条评论下进行的引用回复，此过程需要查询此条回复链接的其它回复，将引用根评论字段加入索引可加速此查询过程；最后，查看某条评论下的子评论，需要查询当前父评论对应的子评论列表，即讲父评论id加入索引以加速此查询。
5. 视频-收藏表中，经常需要查看用户的收藏视频，引入用户id为索引可以加速此过程；另外，考虑到用户注销或视频被删除的可能，将用户id加入外键，以保持一致性。视频-喜欢表、视频-分享表的索引和外键设计与视频-收藏表类似，下面不再赘述。
6. 视频-观看表中，用户经常会查看自己的浏览记录，此过程是以用户id去观看表中查询的，将用户id加入索引会显著提高此查询过程的效率。

3.4.5 推荐算法介绍

3.4.5.1 用户侧基于概率矩阵的推荐算法

动态概率矩阵

用户侧的推荐注重用户本身的视频口味和偏好，本项目设计**动态概率矩阵**用于视频推荐，考虑偏好的种类、偏好的博主、大众化推荐、流量推荐、广告植入，共5种推荐逻辑。5种逻辑的权重分布在本项目中按照经验被设置为[0.4, 0.1, 0.2, 0.1, 0.2]，若是新注册用户，由于缺少其偏好信息，5种逻辑的权重被修改为[0, 0, 0.6, 0.2, 0.2]。

其中，偏好种类是指用户对最近观看的9种分区的视频的喜爱程度，按照观看数、点赞数和收藏数以不同权重计算概率，具体计算公式如下：

$$Type_i = \frac{numType_i}{\sum_{1}^{9} numType_i},\ i\varepsilon\,[1,9]$$

对应到图中第一行的9个概率，简写为tpi。

偏好博主是指用户对最近观看视频对应的上传者（作者）的喜爱程度，计算每个博主的分数，排序后取分数最高的前几位（本项目设置为5位）；然后，归一化分数得到概率向量，即图中第二行的5个概率，记为upi_p。计算公式如下：

$$Up_i\_P = \frac{upCount_i}{\sum_{1}^{5} upCount_i},\ upCount_i = upLiked_i + upFavorite_i + upWatched_i,\ i\varepsilon\,[1,$$

其中，upLiked、upFavorite、upWatched分别指用户点赞、收藏、观看某博主的视频数量。

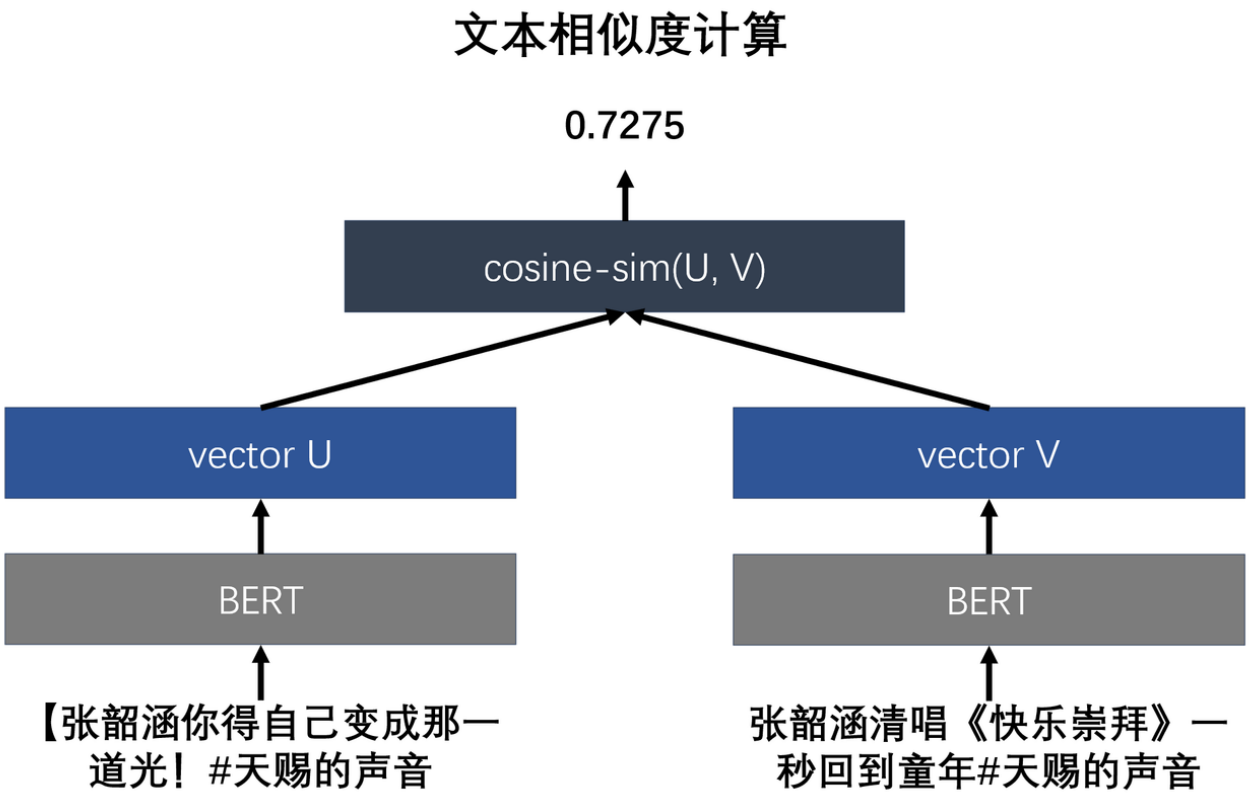
大众化推荐是指整个短视频平台中适用所有用户的观看概率，即根据经验性设置用户对：播放量高、点赞数多、收藏量大、评论数多、分享数多的热门视频均具有不同程度的喜爱，本项目设置这5项热门视频的推荐概率分别为[0.1, 0.4, 0.15, 0.2, 0.15]。

流量推荐是指依据固定的0.1概率为用户推荐被充值购买流量的视频。

广告植入是指依据固定的0.2概率为用户强行推荐广告视频。

以上，便是概率矩阵中包含的5种推荐逻辑，不但实现了对用户的个性化推荐、避免用户陷入信息茧房，同时也兼顾了商业需求平衡。

3.4.5.2 视频侧基于Text2Vec特征相似度的推荐算法



视频侧的推荐算法注重当前播放视频与其它视频的相关性，即相关视频推荐。该算法利用预训练的NLP模型将句子embedding为长为768的向量，然后通过向量即可计算每个句子之间的相似度，可用如下公式表示：

$$Sim_{i,j} = \cos ine\left(BERT\left(sentenceA\right), BERT\left(sentenceB\right)\right), i, j\varepsilon Vids$$

由于每个句子是从视频的标题种中获取的，因此，可以找到某个视频的不同相关程度的视频。本项目中，为每一个视频截取相关性最高的200条视频id，播放该视频时，将会在相关视频栏分批次显示这200条视频。

4 测试

4.1 视频数据来源

本项目开发及测试时获取了视频网站的9900余条视频数据，除了作为基础数据供平台使用外，还为推荐系统的推荐视频生成提供了测试数据。

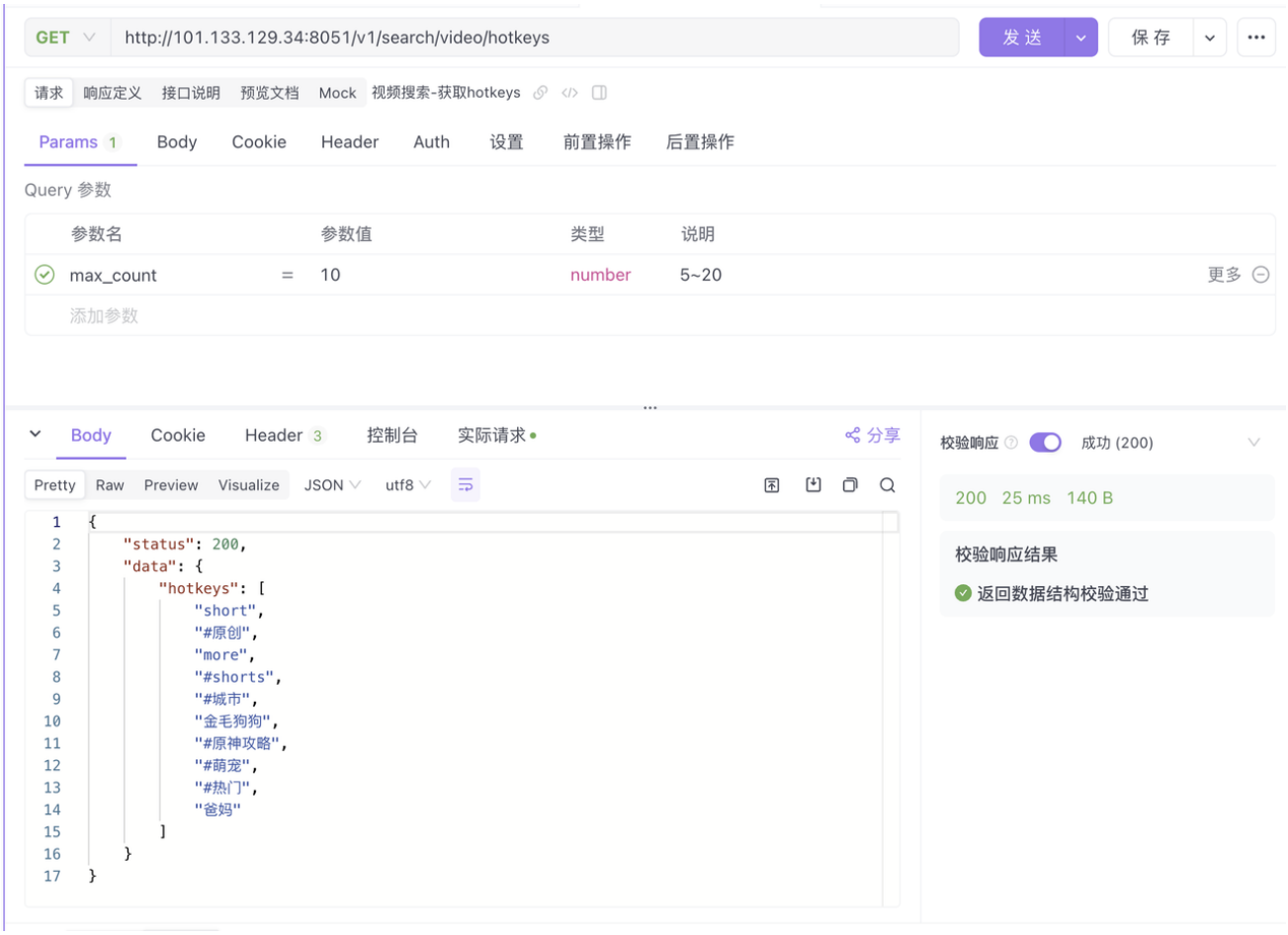
视频数据涵盖了本项目的9个分区：知识、热点、游戏、娱乐、二次元、音乐、美食、体育、时尚，作者包括“CCTV中国中央电视台”、“浙江卫视音乐频道”、“China Focus”、“瑶瑶别追了”、“综艺菌”等热门短视频创作者。

4.2 功能测试

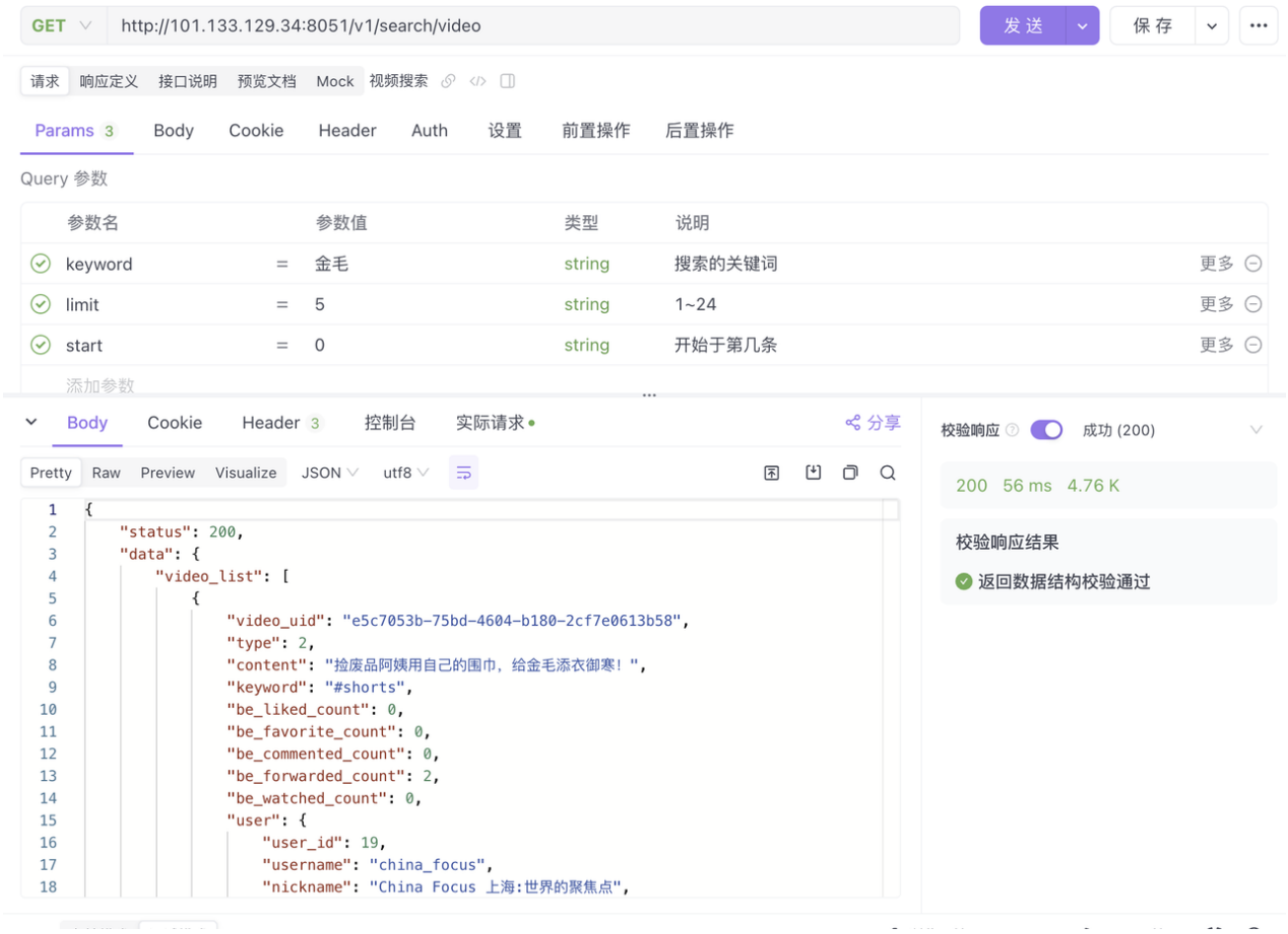
部分接口功能测试示例

由于团队已录制较为完善的功能展示demo视频，故仅在文档中做测试示例。

4.2.1 视频热词获取

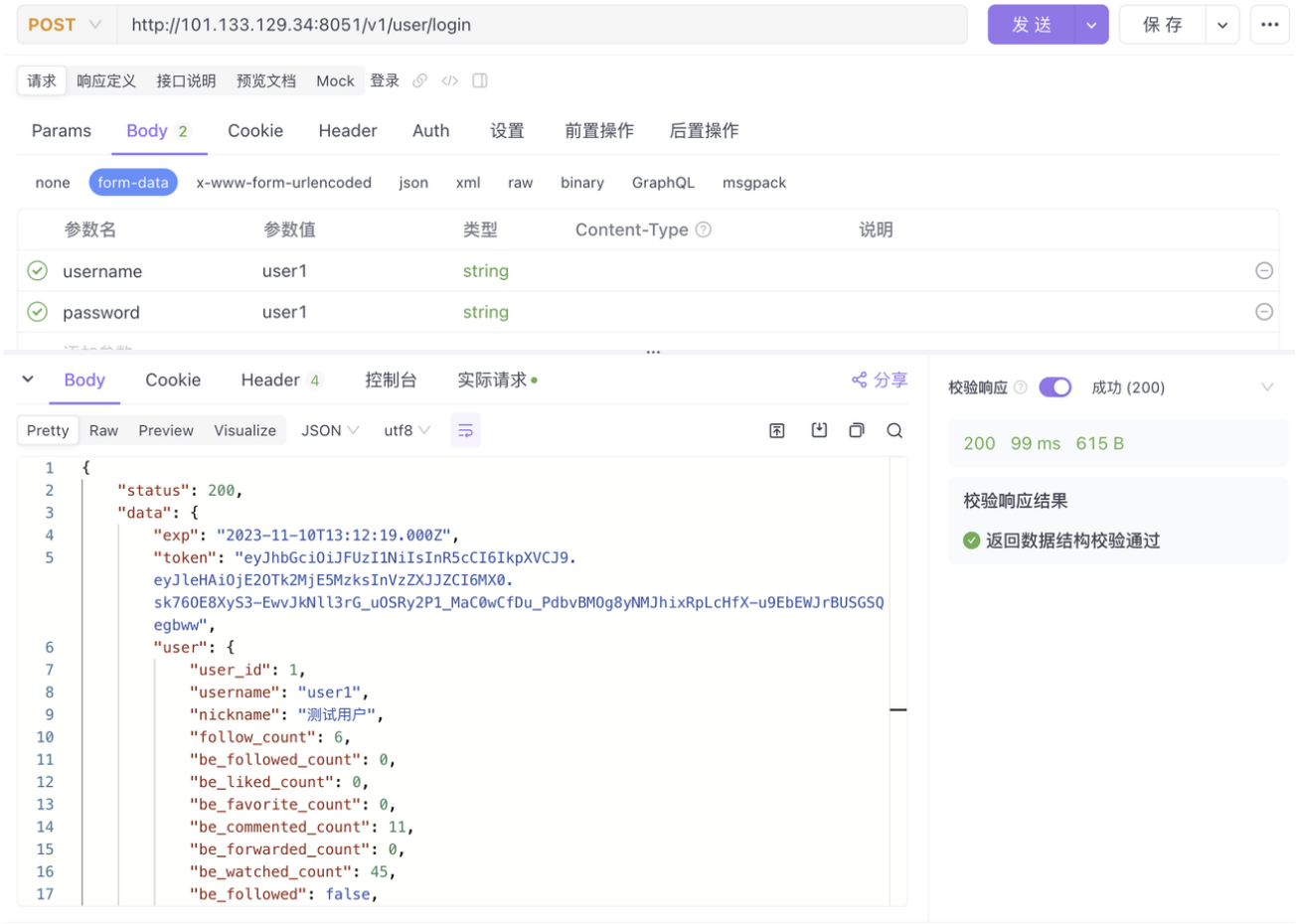


4.2.2 用户搜索视频




```
{
    "status": 200,
    "data": {
        "video_list": [
            {
                "video_uid": "e5c7053b-75bd-4604-b180-2cf7e0613b58",
                "type": 2,
                "content": "捡废品阿姨用自己的围巾，给金毛添衣御寒！",
                "keyword": "#shorts",
                "be_liked_count": 0,
                "be_favorite_count": 0,
                "be_commented_count": 0,
                "be_forwarded_count": 2,
                "be_watched_count": 0,
                "user": {
                    "user_id": 19,
                    "username": "china_focus",
                    "nickname": "China Focus 上海:世界的聚焦点",
```

4.2.3 用户登录



5 视频demo

bilibili链接： https://www.bilibili.com/video/BV14G411Q7CN/

视频名称：ACES短视频DEMO-ACES#JS队